REMARKS

Applicant requests that the docket number of this case be changed to NC 84,834.

Claims 1-48 are pending in the application. No claims are presently allowed.

The paragraph beginning at page 4, line 22 is amended to change a typographical error from "pocket" to "packet."

Claims 1, 3, 17, 19, and 35 are amended to add a missing "and."

Claims 2, 18, and 34 are amended to change "chunks" to "at least one chunk."

Claims 12, 13, 28, 29, 44, and 45 are amended to change "second memory" to "packet memory." The original term was an erroneous combination of "packet memory" and "second type" from the independent claims.

None of the amendments are narrowing amendments.

Claim Rejections – 35 U.S.C. § 112

Claims 2, 18, and 34 have been rejected under 35 U.S.C. § 112, second paragraph as being indefinite for reciting "the chunks" without antecedent basis. By this amendment, this is corrected to "the at least one chunk," which has antecedent basis in claims 1, 17, and 33 respectively.

Claim Rejections – 35 U.S.C. § 102

Claims 1-48 have been rejected under 35 U.S.C. § 102(e) as being anticipated by Bertagna et al., US 6,088,745.

Bertagna discloses a method and apparatus for receiving packets from a FIFO, allocating them to contiguous address spaces in a packet queue, and dispatching them to output ports.

Claim 1 is to an apparatus comprising a buffer memory and a packet memory. The buffer memory stores data organized into at least one chunk based on a linked list. The data is associated with a connection identifier corresponding to a channel in a network. The connection identifier identifies a connection in the channel. The data is part of a data stream associated with the connection. The packet memory is coupled to the buffer memory to provide access to the stored data when a transfer condition occurs.

The Examiner stated that the buffer memory is anticipated by the FIFO and that the data being organized into at least one chunk based on a linked list is anticipated by the data

buffers shown within the packet queue. However, in Bertagna, the linked list is stored only in the packet queue, not in the FIFO. The FIFO appears to contain only constant-byte segments of packets (col. 4, lines 13-14) that are not disclosed as being in a linked list. The linking occurs as this data is placed into the packet queue. Further evidence that the FIFO does not contain the linked list of the packet queue is the fact that segments may not arrive in the FIFO in packet-order (col. 4, lines 14-16), but the chunks (of segments) are stored contiguously in the packet queue. The present claim 1 recites that the linked list must be stored in the buffer memory at some point.

Claim 17 is to a method of storing the linked list of chunks in the buffer memory and providing access to the stored data using a packet memory. As above, Bertagna does not disclose storing the linked list in the buffer memory (FIFO).

Claim 33 is to a system comprising a channel, a data buffer circuit, an input buffer memory, packet memory, and output buffer memory. The input buffer memory stores the linked list. As above, Bertagna does not disclose storing the linked list in the input buffer memory (FIFO).

The Examiner stated that the output memory buffer is anticipated by the output ports. However, Bertagna does not disclose that the output ports are capable of storing data. The ports are a means to dispatch data out of the disclosed system. In the present claim 33, the output memory buffer stores "the data transferred from the input buffer memory," which is a linked list of chunks. Bertagna does not disclose that the linked chunks are transferred to the output ports. Instead, whole packets are sent to the output ports (col. 6, lines 6-24).

Claims 2-16, 18-32, and 34-48 depend from and contain all the limitations of claims 1, 17, and 33 respectively, and asserted to distinguish from the reference in the same manner as claims 1, 17, and 33.

Further, as to claims 8 (9-16 dependent thereon), 24 (25-32 dependent thereon), and 40 (41-48 dependent thereon), these claims recite an ingress queue to buffer a data stream, and a queue segmenter to chunk the data stream. The Examiner stated that the ingress queue is anticipated by the FIFO. However, the Examiner had already stated that the FIFO anticipated the buffer/input buffer memory. The ingress queue buffers the input stream (page 10, line 3), but the buffer/input buffer memory stores the linked list of chunks. The FIFO cannot anticipate both the ingress queue and buffer/input buffer memory, as the FIFO only stores data in one format.

As to claims 10 (11-16 dependent thereon), 26 (27-32 dependent thereon), and 42 (43-48 dependent thereon), these claims recite that the input buffer memory comprises a queue having a threshold and being configured to store the at least one chunk. The Examiner stated that the input buffer memory is anticipated by the FIFO, and that the threshold is anticipated by the filling of buffers 322, 324, etc. However, the buffers 322, 324 are part of the packet queue 120 and not part of the FIFO. The claims recite that the queue is in the input memory buffer. Further, as stated above, the FIFO does not store the chunks.

As to claims 11 (12-16 dependent thereon), 27 (28-32 dependent thereon), and 43 (44-48 dependent thereon), these claims recite that the transfer condition includes at least one of an overflow of the threshold, the packet size, and a scheduled egress request. The Examiner stated that the overflow of the threshold is anticipated by the filling of buffers 322, 324, etc. However, these buffers are in the packet queue 120. The present claims recite that the overflow occurs in the input memory buffer, which the Examiner has stated to correspond to the FIFO.

As to claims 12 (13-16 dependent thereon), 28 (29-32 dependent thereon), and 44 (45-48 dependent thereon), these claims recite a data combiner that combines the data portions of chunks and burst writes the combined chunks to the packet memory or to an output buffer memory, depending on the length of the packet. (The original claim recited "second memory" instead of "packet memory.") The Examiner stated that the write circuit is anticipated by the FIFO queue. However, in Bertagna, the data is written from the FIFO to the packet queue, while adding the chunk headers to the data. In the present claims, the data is written from the input buffer memory to the packet memory, while removing the chunk headers. Only the data portions of the chunks are combined and written to the packet memory. See also Fig. 8B showing packet memory 230 containing combined chunk 810 without the headers, which have been moved to the output buffer memory 240. Further, Bertagna does not disclose the possibility of writing the data from the FIFO directly to an output buffer memory, but only to the packet queue, nor is any output memory buffer disclosed. As explained above regarding claim 33, the output port is not an output buffer memory. Also, Bertagna discloses no method of writing the data to different memories depending on the length of the packet, but only to different addresses with the packet queue.

As to claims 13 (14-16 dependent thereon), 29 (30-32 dependent thereon), and 45 (46-48 dependent thereon), these claims recite a list creator to create an ordered list of pointers to be

transferred to the output buffer memory; a read circuit to burst transfer the contiguous data block from the packet memory to the output buffer memory when the packet size indicates that the packet is long; and an egress queue to buffer the contiguous data block transferred from the output buffer memory. The Examiner stated that the ordered list of pointers to be transferred to the output buffer memory is anticipated by the transfer shown in Fig. 5 of Bertagna. However, this figure shows that only the data portions are transferred to the ports. The headers, shown with connecting, dotted arrows are not transferred. There would be no need to transfer the headers, as they refer to addresses in the packet queue, which would have no relevance to the output port. The present claims require that the pointers themselves are transferred to the output buffer memory. Further, as explained above regarding claim 33, the output port is not an output buffer memory.

As to claims 14 (15 and16 dependent thereon), 30 (31 and 32 dependent thereon), and 46 (47 and 48 dependent thereon), these claims recite that the first type (of memory) is a static random access memory and that the second type (of memory) is a synchronous dynamic random access memory. The Examiner stated that the packet queue in Bertagna may be implemented in random access memory. However, the specific types of random access memory recited in the present claims are nowhere disclosed in Bertagna and are not anticipated.

As to claims 15, 31, and 47, these claims recite that the input buffer memory and output buffer memory have the same size. The Examiner stated that this is anticipated by the standard-sized buffers of col. 2, lines 32-35. However, these standard-sized buffers are in the packet queue, not in the FIFO, which the Examiner has stated to correspond to the input buffer memory, or output ports, which the Examiner has stated to correspond to the output buffer memory.

As to claims 16, 32, and 48, these claims recite that the input buffer memory and output buffer memory have different sizes. The Examiner stated that this is anticipated by col. 2, lines 23-31 where output buffer storage may be implemented in a substantially smaller memory. However, the cited portion is referring to the packet queue, not the FIFO or output ports.

In view of the foregoing, it is submitted that the application is now in condition for allowance.

In the event that a fee is required, please charge the fee to Deposit Account No. 50-0281,

and in the event that there is a credit due, please credit Deposit Account No. 50-0281.

Respectfully submitted,

Joseph T. Grunkemeyer
Reg. No. 46,746
Phone No. 202-404-1556
Office of the Associate Counsel
(Patents), Code 1008.2
Naval Research Laboratory
4555 Overlook Ave, SW.
Washington, DC 20375-5325